

### REMARKS

Claim 1-3, 5-12, and 16-21 are presented for further examination. Claims 1, 8, 11, and 16-20 have been amended. Claims 4 and 13-15 have been canceled, and claim 21 is new.

In the Office Action mailed July 6, 2006, the Examiner asserted that the oath or declaration filed on December 16, 2005, was defective and that a new oath or declaration in compliance with 37 C.F.R. § 1.67(a) identifying the application by application number and filing date was required. Remarks by the Examiner further state that the declaration did not identify the mailing address of each inventor, although the mailing address may be provided in an Application Data Sheet, and that the declaration did not identify the city and state or foreign country of residence of each inventor.

Claims 13-20 were rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter. Claims 1-20 were rejected under 35 U.S.C. § 102(b) as anticipated by newly-cited U.S. Patent No. 5,613,120 ("Palay et al.").

Applicant respectfully disagrees with the bases for the rejections and requests reconsideration and further examination of the claims.

### Declaration

The Examiner has found the new Declaration filed December 16, 2005, to be defective as set forth above. Applicant respectfully disagrees. The new Declaration filed on December 16, 2005, does identify the application by application number and filing date. In addition, the original declaration was filed with an Application Data Sheet stating the inventor's mailing address, which is in accordance with 37 C.F.R. § 1.67(a). If the Examiner still finds the Declaration and the Application Data Sheet to not be in compliance with the Rules, applicant's undersigned representative respectfully requests a telephone conference with the Examiner to expeditiously resolve the matter.

Claims 13-20 Statutory Subject Matter

Claims 13-15 have been canceled. Claims 16-20 have been amended to comply with section 101 requirements in that the computer program product is recited as being implemented in a computer-readable medium (claim 16) or implemented in an electronic processor (claim 19). No new matter has been added. Support for these amendments can be found in the specification at page 4, lines 1-19.

Claim Rejection over Palay et al.

Palay et al., U.S. Patent No. 5,613,120, is directed to a system and method for enabling, without recompilation, modification of class definitions and implementations in an object-oriented computer program. Palay et al. specifically teach that class information in the object files is merged together and duplicate definitions are removed (see column 28, lines 39-61). After this merging step has taken place, the linker evaluates the symbol values only once (see column 29, lines 24-35). In contrast, the disclosed and claimed embodiments of the invention provide a linker that executes the linking and relocation process on the object code modules a number of times in order to output a target executable in the optimum form.

More particularly, the claimed invention records the state of certain variables during each linker pass in order to control and optimize this iterative process. This particular aspect of the invention as set forth in the claims is described in the specification, for example, at pages 16 to 18, where symbol values are calculated a plurality of times. Palay et al. merely merge class information in the object file together and duplicate definitions are removed after which the linker evaluates the symbol values only once.

There is absolutely no disclosure or suggestion of executing a plurality of linker passes and recording the state of variables during each linker pass. Rather, Palay et al. teach that only one linker pass is preferred, and that if the executable and objects are not in the same state, then a run-time linker must redo many of the calculations done by the static linker (see Palay et al., column 30, lines 1-27). There is no teaching or suggestion in Palay et al. of recording the state of certain variables during any additional linker pass in order to control and optimize the output target executable.

Claim 1, which has been amended to include the limitations of dependent claim 4, is directed to a method of linking a plurality of object code modules to form an executable program, the method comprising linking a plurality of object code modules by reading at least one relocation instruction from the set of relocation instructions, and where the relocation instruction is a data retrieval instruction, determining the symbol identified by the symbol field and retrieving one of the plurality of symbol attributes associated with the symbol in dependence on contents of the symbol attributes field of the instruction. Claim 1 further recites recording a pass value indicative of the number of times the set of relocation instructions from the plurality of object code modules have been read.

Nowhere do Palay et al. teach or suggest recording the state of certain variables during any additional linker pass in order to control and optimize the output target executable. The Examiner has pointed to column 28, lines 50-61, in asserting that the subject matter in claims 4, 8, 9, 10, and 16-20 relating to this feature is disclosed. However, column 28, lines 50-61 of Palay et al. do not disclose storing the state of certain variables (such as a pass value) during each linker process. Instead, this portion of Palay et al. only discloses the merging of class information in the object files prior to calculating symbol values. For example, there is no disclosure or suggestion of recording a pass value.

In view of the foregoing, applicant respectfully submits that claim 1 is clearly allowable.

Dependent claims 2, 3, and 5-10 are allowable for the features recited therein as well as for the reasons why claim 1, from which they all ultimately depend, is allowable.

Claim 11 is directed to a method of linking a plurality of object code modules to form an executable program in which, during the linking process, at least one relocation instruction is read, a pass value indicative of the number of times the set of relocation instructions have been read is recorded, and when the relocation instruction is a data instruction, identifying the symbol identified by the symbol field and determining if the associated symbol value has been retrieved by a further data retrieval instruction during the current or previous repetition of the set of relocation instructions, and placing a determined value in a store in response to the determination.

As discussed above, Palay et al. do not disclose or suggest the combination recited in claim 11. For example, nowhere do Palay et al. teach or suggest recording the state of certain variables, such as a pass value indicative of the number of times the set of relocation instructions have been read. Rather, Palay et al. merely disclose storing symbol values. There is no disclosure of the aforementioned step in relation to calculating symbol values. Applicant respectfully submits that claim 11 and dependent claim 12 are allowable.

Independent claim 16 is directed to a computer program product implemented in a computer-readable medium, the product including program code means arranged so that, when run on a computer and during a linking process, at least one relocation instruction is read from the relocation instructions, and, *inter alia*, a path value indicative of the number of times the relocation instructions from the plurality of object code modules have been read is recorded. Similarly, claim 19 is directed to a computer program product implemented in an electronic process that includes program code means arranged so that, when run on a computer, and during the linking process, the program code means reads at least one relocation instruction from a set of relocation instructions, records a pass value indicative of a number of times a set of relocation instructions have been read, and, *inter alia*, places a determined value in a store in response to determination of the symbol value. Applicant respectfully submits that these claims and all claims depending therefrom, *i.e.*, claims 17, 18, 20, and 21, are all allowable for the reasons discussed above with respect to claims 1 and 11.

In view of the foregoing, applicant respectfully submits that all of the claims in this application are in condition for allowance. In the event the Examiner finds minor informalities that can be resolved by telephone conference, the Examiner is urged to contact applicant's undersigned representative by telephone at (206) 622-4900 in order to expeditiously resolve prosecution of this application. Consequently, early and favorable action allowing these claims and passing this case to issuance is respectfully solicited.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

All of the claims remaining in the application are now clearly allowable.  
Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,  
SEED Intellectual Property Law Group PLLC

/E. Russell Tarleton/  
E. Russell Tarleton  
Registration No. 31,800

ERT:jk

701 Fifth Avenue, Suite 6300  
Seattle, Washington 98104-7092  
Phone: (206) 622-4900  
Fax: (206) 682-6031

844555\_1.DOC